Local Structure Matters Most: Perturbation Study in NLU

Louis Clouâtre[1,3] **Prasanna Parthasarathi**[2,3] **Amal Zouaq**[1] **and Sarath Chandar** [1,3,4]
[1] Polytechnique Montréal
[2] School of Computer Science, McGill University
[3] Quebec Artificial Intelligence Institute (Mila)
[4] Canada CIFAR AI Chair

Abstract

Recent research analyzing the sensitivity of natural language understanding models to word-order perturbations has shown that neural models are surprisingly insensitive to the order of words. In this paper, we investigate this phenomenon by developing order-altering perturbations on the order of words, subwords, and characters to analyze their effect on neural models' performance on language understanding tasks. We experiment with measuring the impact of perturbations to the local neighborhood of characters and global position of characters in the perturbed texts and observe that perturbation functions found in prior literature only affect the global ordering while the local ordering remains relatively unperturbed. We empirically show that neural models, invariant of their inductive biases, pretraining scheme, or the choice of tokenization, mostly rely on the local structure of text to build understanding and make limited use of the global structure.

1 Introduction

Recent research has shown that neural language models have an understanding of well-formed English syntax in recurrent neural networks, convolutional neural networks, and in large pretrained (PT) Transformers (Gulordava et al., 2018; Zhang and Bowman, 2018; Chrupała and Alishahi, 2019; Lin et al., 2019a; Belinkov and Glass, 2019; Liu et al., 2019a; Jawahar et al., 2019; Rogers et al., 2020). Other studies, however, take a critical stance with experiments suggesting that models may be insensitive to word-order perturbations (Pham et al., 2021; Sinha et al., 2021, 2020; Gupta et al., 2021; O'Connor and Andreas, 2021), showing that shuffled word-order has little to no impact during training or inference with neural language models. While some research show that models learn some abstract notion of syntax, further probing into their insensitivity to the perturbation of syntax is necessary. Specifically, *What are the underlying mechanisms causing those unintuitive, or unnatural, results from neural models* is still a largely unanswered question.

Recent research exploring the sensitivity to syntax of pretrained models has primarily been applying perturbations to text through perturbing the order of words (Pham et al., 2021; Sinha et al., 2021, 2020; Gupta et al., 2021; O'Connor and Andreas, 2021). Perturbations applied and quantified at this granularity of text offer only a limited understanding of the learning dynamics of the neural language models. Analyzing perturbations at a finer granularity such as subwords (Bojanowski et al., 2017) or characters (Gao et al., 2018; Ebrahimi et al., 2018), may provide a deeper insight into the insensitivity to word-order of neural models.

In this paper, we define two types of structure[1] in text, global which relates to the absolute position of characters, and local, which relates to the relative position of characters to their immediate neighbors. We observe from our experiments (§ 5) that most perturbations proposed and analyzed in the literature will perturb the global structure with different reordering of words, while the amount of disturbance to the local structure remains limited. *We hypothesize that the local structure, more so than the global structure, is necessary for understanding in natural language tasks.* By applying perturbations of varying degrees to the local structure, while controlling for the amount of global perturbations, we are able to measure how essential it is to a neural model understanding of text. We demonstrate the sensitivity to local structure of model performances in English natural language understanding (NLU) (GLUE (Wang et al., 2019a)) and their relative insensitivity to the global structure, and control for many potential confounding factors that would otherwise provide an alternative explanation to our results.

[1] Structure here relates to the organization of characters in the text.

Findings of the Association for Computational Linguistics: ACL 2022, pages 3712 - 3731
May 22-27, 2022 ©2022 Association for Computational Linguistics

Our contributions are as follows:

- We show that the performance of neural models – Transformers and others, pretrained or not – on perturbed input strongly correlates with the amount of preserved local structure of text.

- We identify possible confounding factors for this phenomenon and construct experiments controlling for them.

- We provide analysis on implications derived from our large array of empirical findings.

2 Related Work

Importance of Syntax Discussions on semantics (Culbertson and Adger, 2014; Futrell et al., 2020) agree on specific orders of words to be necessary for comprehending text. Psycholinguistic research (Hale, 2017) corroborates this through evaluating sentence comprehension mechanisms of humans. Hence, interpreting language as a bag-of-words could limit the expressions conveyed through the word-orders (Harris, 1954; Le and Mikolov, 2014) and understanding syntax[2] becomes an essential artifact. Recently, Mollica et al. (2020) found that humans were robust to word-ordering perturbations in text as long as local ordering of text was roughly preserved.

Prior works have explored the relationship between neural models and syntax. Goldberg (2019); Hewitt and Manning (2019) both show that BERT (Devlin et al., 2019) models have some syntactic capacity. Lin et al. (2019b) show that BERT represents information hierarchically and concludes that BERT models linguistically relevant aspects in a hierarchical structure. Tenney et al. (2019); Liu et al. (2019b) show that the contextual embeddings that BERT outputs contain syntactic information that could be used in downstream tasks.

While it seems that syntax is both important, and to an extent, understood by the recent family of PT models, it is unclear how much use they make of it. Glavaš and Vulić (2020) showed that pretraining BERT on syntax does not seem to improve downstream performance much. Warstadt et al. (2020) showed that while models such as BERT do understand syntax, they often prefer not to use that information to solve tasks. Ettinger (2020); Pham et al. (2019); Sinha et al. (2020); Gupta et al. (2021) show that large language models are insensitive to minor perturbations highlighting the lack of syntactic knowledge used in syntax rich NLP tasks. Sinha et al. (2021) show that pretraining models on perturbed inputs still obtain reasonable results on downstream tasks, showing that models that have never been trained on well-formed syntax can obtain results that are close to their peers.

While syntactic information seems vital to language, and large PT models seem to be at least aware of syntax, the lack of sensitivity of neural models to perturbation of syntax motivates further probing.

Text Perturbations Several different types of reordering perturbation functions and schemes have been explored to understand and study neural architectures' (in)sensitivity to word-order. The class of perturbation analysis could broadly be split into three categories: deletion, paraphrase injection, and reordering of tokens. Sankar et al. (2019) explore utterance and word-level perturbations applied to generative dialogue models to highlight their insensitivity to the order of conversational history. On natural language classification tasks, Pham et al. (2021) define n-grams for different values of n and shuffle them to highlight the insensitivity of PT models. They show that shuffling larger n-grams has a lesser effect than shuffling smaller n-grams, suggesting that preserving more local structure causes less performance degradation. Studying textual entailment tasks, Sinha et al. (2020) perform perturbations on the position of the words, with the criteria that no word remains in its initial position.

Hsieh et al. (2019) propose a suite of adversarial attacks that replace one word in the input to cause a model to flip its correct prediction. Gupta et al. (2021) combine several types of destructive transformations — such as sorting, reversing, shuffling words — towards removing all informative signals in a text. Along similar lines, Wang et al. (2019b) inject noise by reordering or deleting articles towards injecting artificial noise to measure the robustness of PT language models. Character-level perturbations that perform minimal flips to cause a degenerate response have been explored by Ebrahimi et al. (2018); Gao et al. (2018). Gao et al. (2018) quantify the perturbation in Levenshtein distance and draw a correlation to the model's perfor-

[2] Preference to a specific word-order over the other and the preference complying with the choices of an average human speaking that language.

mance. This work is closely related to our own. We demonstrate that our hypothesis, the importance of local ordering, is a much more robust explanation of the degradation in performance of models than the Levenshtein distance.

Quantifying Perturbations Several popular similarity metrics can be used to measure perturbations. Metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) will treat text as a sequence of words, from which a measure of overlap is computed. The Levenshtein distance (Levenshtein, 1966; Yujian and Bo, 2007), or the *edit* distance, measures the minimum amount of single-character edits (insertions, deletions, or substitutions) necessary to match two strings together. In the context a shuffling text, it will roughly count the amount of characters that have been displaced. Parthasarathi et al. (2021) observed that learned metrics like BERT-Score (Zhang et al., 2019) and BLEURT (Sellam et al., 2020) are often unaffected by minor perturbations in text which limits their usefulness in measuring perturbations. Character-level metrics, such as the character n-gram F-score (chrF) (Popović, 2015) offer a character-aware approach to measuring similarity of n-gram overlap between two texts. In the context of shuffling this, this will represent roughly the amount of character n-gram that have been changed by the shuffling.

3 Measuring Local and Global Pertubations

To properly analyze different perturbations to the local and global structure of text, we first require a way to measure perturbations to said structures. The *global* structure here relates to the absolute position of characters in a text, and the *local* structure relates to the neighboring character of any other character in a text.

3.1 Character bigram F-score (chrF-2)

To measure local perturbations, we use the chrF (Popović, 2015) metric. chrF is an n-gram overlap metric that is applied to characters. The goal here is to isolate the smallest unit of local structure that we can quantify, character 2-grams being preserved after perturbations. We therefore use a minimal and maximal n-gram length of 2. We use the default β value of 3. Our metric is equivalent to calculating the F3-score of character 2-gram overlap between the unperturbed text and the perturbed text, taking whitespaces into account.

3.2 IDC

To measure the global perturbations, we introduce the **I**ndex **D**isplacement **C**ount (IDC) metric, which measures the average distance traversed by every character after perturbations.

Let a string, $x_i = (c)_k^i$, be denoted by a sequence of characters $c_0, \ldots, c_k$, where k is the length of the string in characters and p^{x_i} denote the positions of characters in x_i. Let $\eta(\cdot)$ be a perturbation operation.

$$x_i' \leftarrow \eta\left(x_i\right), \tag{1}$$

where x_i' denote the perturbed string with positions of the characters specified by $p^{x_i'}$.

$$IDC \leftarrow \frac{1}{k^2} \sum_{j=1}^{k} \left\| p^{x_i'}(j) - p^{x_i}(j) \right\|_1 \tag{2}$$

The denominator k^2 normalizes the average by the length of the text[3]. Intuitively, an IDC of 0.3 would imply that characters in the perturbed text have moved 30% of the text length on average. The values of IDC will lie in the range $[0, 0.5]$, where 0.5 would be obtained by reversing a text at the character level.

3.3 Compression Rate (Comp)

Finally, to measure local perturbations to words and subwords, we could count the rate of out-of-vocabulary (OOV) tokens introduced by the perturbations. As our experiments make use of a subword vocabulary (Sennrich et al., 2015) which can represent any string of English characters without OOV tokens, the compression rate (Xue et al., 2021), as measured by the length of the original string in characters divided by the length of the tokenized string, will serve as a proxy to measuring OOV tokens. As more local perturbations are applied, more and more subwords will be broken into smaller subwords which will yield a lesser compression of text through tokenization. The tokenizer of the RoBERTa-Base model (Liu et al., 2019c) is used to calculate the compression rate in all cases.

4 Perturbation Functions

Towards conducting a detailed analysis on the effect of perturbations on the performance of neural language models, we define three granularities

[3] k^2 is used to normalize as we sum k times a number that is between 0 and k, where k is the text length.

of perturbation functions — *word-level*, *subword-level* and *character-level*. The subwords are taken from the RoBERTa-Base vocabulary. We define the perturbation functions as generic operations that can be applied across the different levels of granularity[4].

Full Shuffle randomly shuffles the position of every word, sub-word, or character, according to the level it is applied to. This transformation should cause a great amount of perturbation to the *global* and *local* structure for the specific granularity.

The scholar is typesetting.

scholar typesetting is The.

Figure 1: Example for word-level full shuffling. The perturbed sentence has a IDC of 0.29 and a chrF-2 of 0.92.

Phrase Shuffle creates chunks of contiguous tokens of variable length, controlled by a parameter ρ, and shuffles the phrases of word, subword, or characters. This perturbation has, on average, the same impact as the full shuffling on the *global* structure as the absolute positions of characters tend to change just as much as full shuffling while preserving a controllable amount of *local* structure.

The scholar is typesetting.

is typeThe schosetting lar.

Figure 2: Subword-level phrase shuffling. The perturbed sentence has an IDC of 0.35 and a chrF-2 of 0.84.

To randomly define our phrases, we traverse the text sequentially on the desired granularity. The entire text is assumed as a single large phrase and is truncated at a token with probability ρ into smaller phrases.

A lower value of ρ leads to longer on average phrases, thus preserving more of the *local* structure while destroying roughly the same amount of *global* structure. In the extreme case with $\rho = 1.0$, phrase shuffling will be equivalent to full shuffling as phrases will all be one token long.

Neighbor Flip Perturbations flip tokens of the chosen granularity with the immediate right neighbor with probability, ρ. This function has, on average, a smaller impact on the *global* structure, as the absolute positions of tokens do not change much but can have an arbitrary large effect on disturbing the *local* structure.

The scholar is typesetting.

heT cshlori sa typeesttnig.

Figure 3: Character-level neighbor flip. The perturbed sentence has an IDC of 0.04 and a chrF-2 of 0.32. Due to a greater distortion to the local order, the model has a greater chance to be sensitive to this perturbation.

The perturbation is applied by traversing the string from left-to-right on the desired granularity and, with a probability ρ, switching the current attended token with the following token. The lower the ρ is, the less perturbation happens, thus preserving more of the *local* structure. This transformation has a limited impact on the *global* metric, thus letting us isolate the impact of perturbations to the different structures.

5 Experiments

5.1 Dataset

We experiment with the GLUE Benchmark (Wang et al., 2019a) datasets, a popular NLU benchmark. We create perturbed versions of the validation set for all tasks with the different perturbation functions defined in § 4. In total, 50 different variations of our perturbation functions are applied by varying the granularity as well as the ρ values, including an unperturbed benchmark version[5].

5.2 Confounding Variables

We have identified several confounding variables that we will attempt to control for in our experimental setup.

Inductive Biases of the neural architecture may yield models that rely on different types of structure. Intuitively, it may be that Transformer-based models, through global self-attention, rely more on global structure than ConvNets which are limited to local information.

[4] Pseudo-code and examples for all perturbations are shown in Appendix B.

[5] The hyperparameters used for the perturbation functions are detailed in Appendix A.

Pretraining may have a large impact on the level of sensitivity to different types of structure. It may be that global structure simply requires more training to be understood and that pretrained models leverage it to a much higher degree than non-pretrained (NPT) models. The specific method used for pretraining may also impact the sensitivity to different types of structures, such as adding permutations to the pretraining objectives.

Tokenization schemes may be the most significant confounding variable. By perturbing the local ordering of characters, we also perturb the vocabulary of models that rely on the precise order of characters.

5.3 Models

We experiment with BiLSTMs (Schuster and Paliwal, 1997), Transformers (Vaswani et al., 2017), and ConvNets to have an appropriate breadth of neural inductive biases. We experiment with three flavor of PT Transformers (RoBERTa-Base (Liu et al., 2019c), BART-Base (Lewis et al., 2019) and CharBERT-Base (Ma et al., 2020)), and a NPT Transformer (RoBERTa-Base architecture) to verify the impact of pretraining. We also experiment with different tokenization schemes, using byte-pair encoding (BiLSTMs, ConvNet, RoBERTa-Base, BART-Base, NPT Transformer) as well as character-level tokenization (BiLSTMs, ConvNet, CharBERT-Base (Ma et al., 2020)), to isolate the impact of the destruction of a model's vocabulary.

The tokenization for PT Transformer models use their corresponding vocabulary, while NPT models (BiLSTM, ConvNet, Transformer) use the RoBERTa-Base vocabulary and the character-level models use characters exclusively as vocabulary[6]. Training is done once on the unperturbed dataset until convergence and evaluation is done on the perturbed version of the validation datasets. The training details can be found in Appendix A.

6 Analysis

6.1 Metrics and GLUE Performance

We compute the average GLUE score of different models applied to the validation data perturbed with our different perturbation functions. The PT RoBERTa-Base results are plotted in Figure 4[7].

First, we observe that word and subword-level perturbations are very limited in their impact on the local structure, but can affect the whole spectrum of global structure. We observe the general trend that the chrF-2 metric strongly correlates with neural models' loss in performance on the GLUE benchmark tasks across all perturbations and granularity of perturbations. While the IDC metric correlates somewhat with performance, it fails to distinguish between neighbor flipping perturbations and phrase shuffle perturbations. The compression rate is strongly correlated with performance on character-level perturbations but does not hold explanatory power for word and subword-level perturbations, as they do not affect the vocabulary, leading to the overall lower rank correlation with performance degradation.

By computing the rank correlation between the GLUE score of the different models on the perturbed samples and the metric measuring the perturbations (Figure 5), we see that the correlation of GLUE score with the chrF-2 metric holds for every single architecture and setting tested. On the other hand, the IDC metric is only weakly correlated with performance decay. This implies that local structure, more so than global structure, is necessary for models to perform NLU. A model being evaluated on a perturbed text with a chrF-2 of 0.7 can be assumed to have much lower performance than on a perturbed text with a chrF-2 of 0.95, irrespective of the granularity or the type of perturbations that yielded those metrics. This is not true of any of the other metrics.

Looking at the individual tasks more closely, as in Figure 6, we see that the conclusions regarding the overall GLUE benchmark do hold for every task individually.

6.2 Effect of Perturbations on Metrics

As intended, the different perturbations have different impact on our metrics, as shown in Figure 4. Thee neighbor flip perturbations objective was to obtain an arbitrary amount of local perturbation for a relatively small amount of global perturbation. We can observe that the IDC metric, which measures the impact to the global structure, is smaller for the neighbor flip than for the phrase shuffle, even when the amount of local perturbation, as measured by the chrF-2 metric, is roughly equivalent. The compression rate is closely tied to the measure of local structure on character-level perturbations,

[6] The CharBERT model uses a mix of characters and subword vocabulary.

[7] Results for all individual models can be found in Appendix C

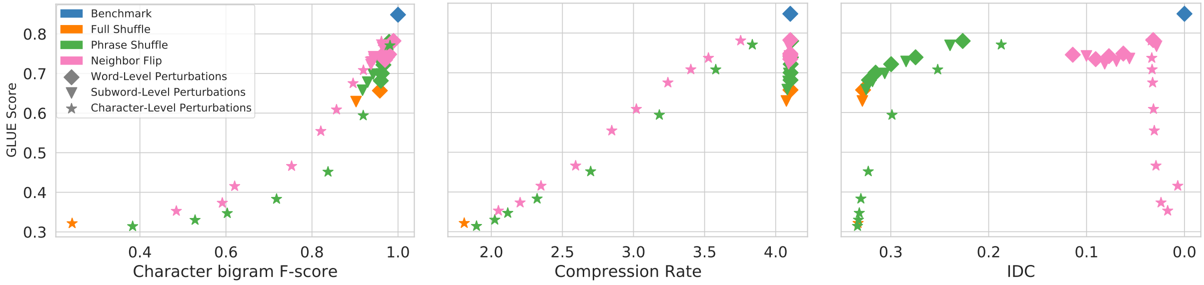


Figure 4: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of PT RoBERTa-Base model tested on the perturbed data. Left is more perturbed, up is better performance. The X-axis of the IDC metric is inverted for clearer comparison.

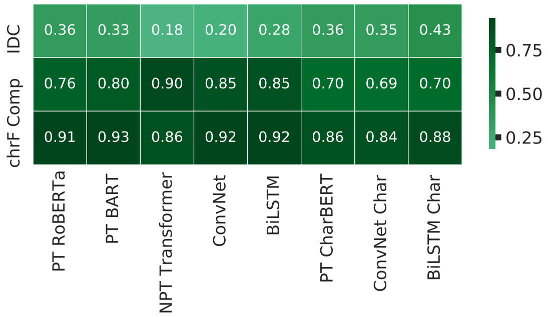


Figure 5: Rank correlation matrix between the models' performance to perturbed samples on the GLUE benchmark and the perturbation quantified by the different metrics. The higher the value the better the metric explains the degradation in performance.

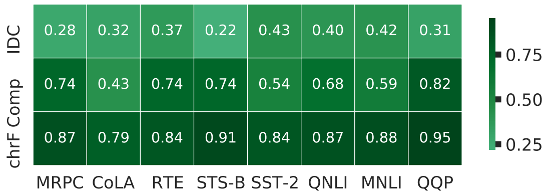


Figure 6: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the PT RoBERTa model.

but is static for word and subword perturbations as the tokens are never impacted.

6.3 Correlation between metrics

To confirm that the chrF-2 metric and the IDC metric do measure orthogonal aspects of structure, we compute their pairwise pearson correlation in the GLUE validation set in Figure 7[8] We also include the compression rate. Specifically, for every sample in the validation set of the GLUE tasks, we perturb them using the different perturbation functions and compute their scores with the different metrics.

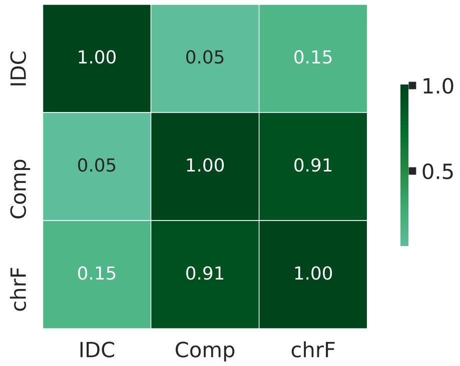


Figure 7: Correlation matrix between the different metrics on the GLUE tasks.

We observe that chrF-2 and IDC have a fairly low correlation, suggesting that the metrics measure different aspects of the perturbations. We also observe a very high correlation between the chrF-2 measure and the compression rate, which motivates experiments that perturb one without impacting the other to isolate the main component causing the performance degradation.

6.4 Model specific analysis

The loss in performance of models in GLUE tasks shows a greater degree of correlation with the chrF-2 metric than any other metric, as shown in Figure 5, with the exception of the NPT Transformer which we discuss in § 6.4.2.

6.4.1 Pretrained vs Non-Pretrained models

Figure 5 demonstrates that perturbations to the local structure explain much of the degradation in performance for both PT and NPT models. De-

[8] For every correlation, we inverted the value of the IDC metric by flipping its signs to make the comparison of the different correlations more straightforward. It is a measure of perturbation and not similarity and is therefore inversely correlated to the GLUE score and the other metrics.

spite the different pretraining schemes used, the PT RoBERTa and BART model have a comparable level of degradation across the different perturbations, showing that the choice of pretraining scheme has a relatively small impact on perturbation resistance.

All NPT models exhibit a strong correlation between the chrF-2 metric and their degradation in performance on the GLUE tasks, which indicates that the sensitivity to local structure is not an artifact of pretraining.

6.4.2 NPT Transformer and Positional Embeddings

Interestingly, the NPT Transformer bucks the overall trend by having very little correlation between its performance and IDC and being more correlated to the compression rate than to the chrF-2 metric. As IDC will roughly measure the distance traversed by characters from their initial position, it having little correlation with performance in NPT Transformers implies that the absolute position of tokens is not taken into account by the NPT Transformers. We hypothesize that learning the positional embeddings requires much more data than is present in a single NLU task, leading the NPT model to act as a bag-of-words model. This would explain why perturbations to the vocabulary are so impactful to the NPT Transformer, as it is unable to correct minor disturbances in words with the context of neighboring words.

Towards studying this, we conduct an ablation study on the impact of positional embeddings with NPT and PT Transformers. We freeze the weights of the positional embeddings to 0, making them have no contribution to the overall output of the model. As we are interested in the marginal utility of positional embeddings with relation to NPT Transformers, we report the difference in performance between the model that has access to those embeddings and the model that does not (Δ GLUE Score). Without positional embeddings, a model has no information on the relative position of inputs and is forced to use only the bag-of-word information. In Figure 8, we can see that the performance of the NPT Transformer without positional embedding varies about $\pm 2\%$, consistent across all levels of perturbations, while the PT model performance is strongly improved by the presence of the positional embeddings. This suggests that NPT Transformers barely make any use of the positional embeddings on those tasks[9].

6.5 Character-Level Experimentation

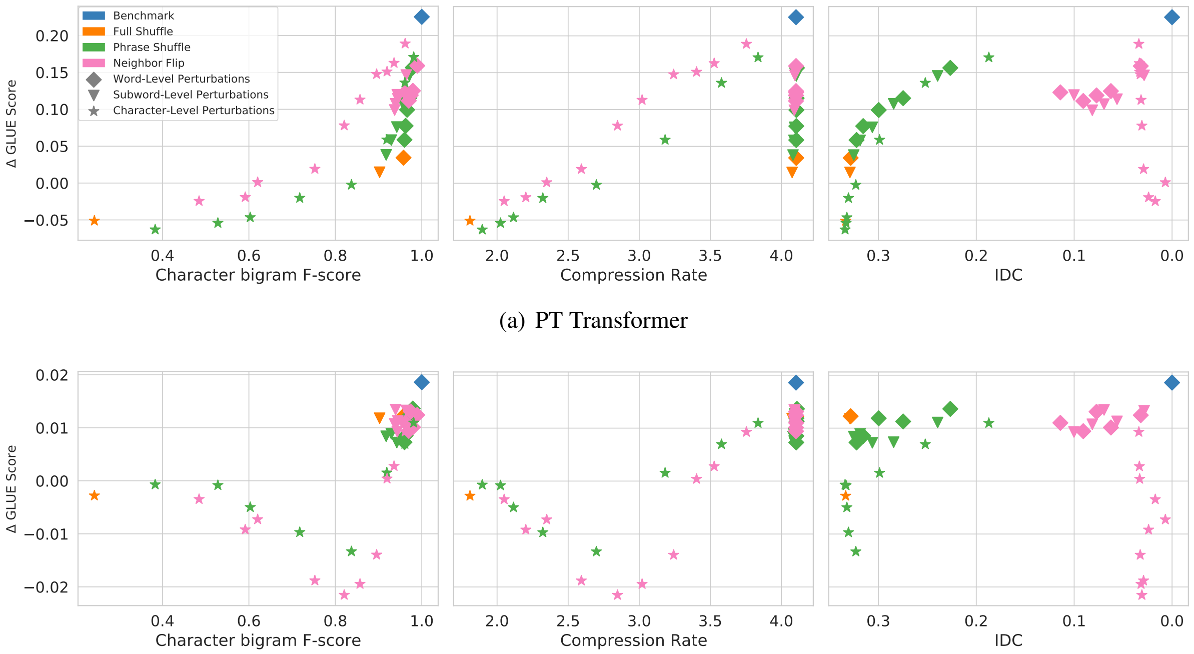
As the results presented from experiments so far use subword tokenization, it is possible that the local perturbations being directly correlated with performance decay could be caused by the perturbation to the vocabulary. To control for vocabulary destruction as a possible explanation for the observed phenomenon, we train character-level BiLSTMs, ConvNets and finetune a PT CharBERT model on all tasks to evaluate whether the correlations between metrics and performance hold without multi-character vocabulary. Results shown in Figure 5 demonstrate that even when using a single-character vocabulary, the correlations between performance for ConvNets, BiLSTMs, and PT Transformers remains roughly static. This implies that the destruction of the specific tokens used by the model is not the main driver for the degradation in performance leaving perturbation to the local structure as the most likely explanation.

7 Discussion

Significance of Results While our results at the extremes may be trivial, such that completely shuffling the order of characters of a text removes all the structure necessary for understanding, and that destroying the local structure to an extreme also prohibits models from building a useful representation of the text, it is not trivial that performance correlates to this degree to local structure across the whole spectrum of perturbations. In Figure 4, fully shuffling the subwords of a text and randomly flipping characters with their neighboring character 10% of the time obtains roughly the same GLUE score and chrF-2 metric despite much different perturbations being applied and much different IDC and compression rate. The removal of any amount of local structure correlating directly to an equivalent drop in performance, with little concern for the granularity or mechanics of that removal of local structure, allows us to make interesting conclusions on the kind of structure that is used by neural models to build understanding.

Adversarial Attacks By better understanding the specific mechanics that can induce failure in neural language models, it is possible to develop models that are more resistant to adversarial attacks.

[9] Further analysis is presented in Appendix C.2



(a) PT Transformer

(b) NPT Transformer

Figure 8: Difference in GLUE scores between a Transformer and the same Transformer trained and tested with positional embeddings frozen at 0. Results for NPT and PT models are shown.

As current models performances can be directly related to the preservation of character 2-grams in all studied variations, this study demonstrates a very likely vector of adversarial attacks that may be important to explore further. Gao et al. (2018) use the Levensthein distance to measure and limit perturbations of black box adversarial attacks, similar research relying on chrF-2 instead may be interesting.

Tokenization Our results on the importance of local structure could bear some implications for tokenization. Recent research trends (Xu et al., 2021; Clark et al., 2021) look at alternatives and improvements to BPE. The current research appears to be pushing towards smaller vocabulary at finer granularity, even exploring simple byte-level representations (Xue et al., 2021; Tay et al., 2021).

We find that local clumps of characters contain the most essential structural information required to solve several NLU problems. As a large part of the complexity of NLU seems to be contained within the meaning of the specific order of clumps of characters, by having more of that local structure fixed through tokenization, it is possible to inject additional useful inductive biases into the model. The perturbation analysis discussed in our work could be used for better construction of vocabulary with improved heuristics.

8 Conclusion

Our results on the relative importance of local structure in relation to global structure hint at the possibility that much of the tested NLU tasks can be solved with a bag-of-words formulation. Intuitively, local structure mainly relates to building meaningful words from the characters of a text whereas the global structure relates to the general order and word-level syntax being maintained. From our experiments, we observe that as long as the local structure is roughly maintained, a majority of NLU tasks can be solved without requiring the global structure. This correlates with similar findings by O'Connor and Andreas (2021). In essence, the structure required to build words seems to be necessary, but much of NLU can be solved with the information of which words (or subwords) are present in the text, without regard to their relative positions.

In this work, we have provided empirical results demonstrating that, for deep learning models in English NLU, perturbations to the local structure, as measured by the chrF-2 metric, is highly correlated to downstream model performance which implies that much of the information obtained from the structure of text comes from the local structure.

Perturbations to the global structure, as measured by IDC, seems to only have a limited correlation to performance, implying that models don't generally rely on it to build understanding. Reflecting on our results, we observe that perturbations on a local level explains the (in)sensitivity of neural language models to perturbations at different granularities on a variety of NLU tasks. This paper hopefully provides useful intuitions on the importance of different types of structures in text for researchers looking into tokenization, neural architectures and adversarial attacks. Although the paper primarily focuses on the effects of perturbations on English texts, extending the study to neural models on other languages will be beneficial.

Acknowledgements

We thank Saujas Vaduguru for the useful comments and discussions on early drafts. This research was supported by Apogée Canada, Canada First Research Excellence Fund program and École Polytechnique Startup Fund PIED. SC is supported by a Canada CIFAR AI Chair and an NSERC Discovery Grant.

References

Eneko Agirre, Llu'is M'arquez, and Richard Wicentowski, editors. 2007. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Association for Computational Linguistics, Prague, Czech Republic.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Grzegorz Chrupała and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Florence, Italy. Association for Computational Linguistics.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. Canine: Pre-training an efficient tokenization-free encoder for language representation. *arXiv preprint arXiv:2103.06874*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Jennifer Culbertson and David Adger. 2014. Language learners privilege structured meaning over surface frequency. *Proceedings of the National Academy of Sciences*, 111(16):5842–5847.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*.

J. Ebrahimi, Anyi Rao, Daniel Lowd, and D. Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *ACL*.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Richard Futrell, Roger P Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2020. Is supervised syntactic parsing beneficial for language understanding? an empirical investigation.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *NAACL-HLT*.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding. *arXiv preprint arXiv:2101.03453*.

John Hale. 2017. Models of human sentence comprehension in computational psycholinguistics. *Oxford Research Encyclopedia of Linguistics*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529, Florence, Italy. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019a. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019b. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *NAACL*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019b. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Pa-

pers), pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Charbert: Character-aware pre-trained language model. In *COLING*.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T. Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. 2020. Composition is the Core Driver of the Language-selective Network. *Neurobiology of Language*, 1(1):104–134.

Joe O'Connor and Jacob Andreas. 2021. What context features can transformer language models use? In *ACL/IJCNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Prasanna Parthasarathi, Koustuv Sinha, Joelle Pineau, and Adina Williams. 2021. Sometimes we want translationese. *arXiv preprint arXiv:2104.07623*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

Thang M. Pham, Trung Bui, Long Mai, and Anh M Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *ArXiv*, abs/2012.15180.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2020. Unnatural language inference. *arXiv preprint arXiv:2101.00010*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, and Dong Yu. 2019b. Improving pre-trained multilingual model with vocabulary expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 316–327, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the Association for Computational Linguistics*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv preprint arXiv:2105.13626*.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. *CoRR*, abs/1504.05070.

A Experiment Details

Model Hyperparameters The results in the paper are averaged over 5 random seeds. We train 5 individual model on all tasks and apply a different random seed to the perturbations to each trained model once. Early stopping was performed after 2 full epochs not resulting in better results on the validation set. All models had similar model sizes, containing between 100 million and 130 million parameters. The ConvNet architecture is the one described in Collobert and Weston (2008) and the BiLSTM architecture is the one described in Zhao et al. (2015). The character embedding ConvNet uses a kernel of size 12 instead of 3, to offset the much longer character sequences. Both the ConvNet and BiLSTM use the same hidden size, dropout and word embedding size as the RoBERTa-Base model. Pretrained models used a learning rate of 2e-5, a batch size of 32, a maximum of 5 epochs and a weight decay of 0.1. Non-pretrained models used a learning rate of 1e-4, a batch size of 128, a maximum of 50 epochs and a weight decay of 1e-6. All experiments used a warmup ratio of 0.06, as described in Liu et al. (2019c). Experiments using characters as input used a maximum sequence length of 2048 inputs. All other experiments used a maximum sequence length of 512. The Winograd Schema Challenge (WNLI) task was omitted from all experiments as it contains well known issues and is often omitted (Liu et al., 2019c; Devlin et al., 2019; Radford and Narasimhan, 2018). The validation set, instead of the test set, is used as the test set is kept private for the GLUE benchmark.

Perturbations Subword-level perturbations were all done with the RoBERTa-Base tokenization. On all level of granularity, we perform one experiment with in the full shuffling setting. On the word and subword-level perturbations we perform phrase-shuffling with ρ values of: [0.8, 0.65, 0.5, 0.35, 0.2] and neighbour-flip shuffling with ρ values of: [0.8, 0.6, 0.5, 0.4, 0.2]. On the character-level perturbations we perform phrase-shuffling with ρ values of: [0.975, 0.95, 0.9, 0.8, 0.65, 0.5, 0.4, 0.3, 0.2, 0.15, 0.1, 0.075, 0.05] and neighbour-flip shuffling with ρ values of: [0.8, 0.65, 0.5, 0.4, 0.3, 0.2, 0.1, 0.075, 0.05, 0.035, 0.025, 0.01]. A total of 11 word-level experiments, 11 subword-level experiments, 27 character-level experiments and the unperturbed benchmark are evaluated for a grand total of 50 different perturbation settings.

B Pseudocode for Metric and Perturbations

```
Function PhrasePerturbation($\rho \leftarrow 0.5$, text←list):
    all_phrases ← list();
    phrase ← list(text[0])
    for token in text[1:] do
        p ∼ Unif([0,1]);
        if $p < \rho$ then
            all_phrases.append(phrase);
            phrase ← list(token)
        else
            phrase ← [phrase, token];
        end
    end
    all_phrases.append(phrase);
    perturbed_text ← ‘’.join(shuffle(all_phrases))
return perturbed_text
```

Algorithm 1: Pseudocode for PhraseShuffle.

```
Function NeighborFlip($\rho \leftarrow 0.5$, $text \leftarrow$ list):
    perturbed_tokens ← list();
    held_token ← list(text[0])
    for token in text[1 :] do
        p ~ $Unif([0, 1])$;
        if $p < \rho$ then
            perturbed_tokens.append(held_token);
            held_token ← list(token)
        else
            perturbed_tokens ← [perturbed_tokens, token];
        end
    end
    perturbed_tokens.append(held_token);
    perturbed_text ← ‘’.join(perturbed_tokens)
return perturbed_text
```

Algorithm 2: Pseudocode for NeighborFlip.

C Other Results

In this section, we add for all other tested models the results that were presented for the RoBERTa-Base model. They were not included in the main paper for simple economy of space.

C.1 PT BART

The PT BART model has results that are very much inline with the PT RoBERTa model.

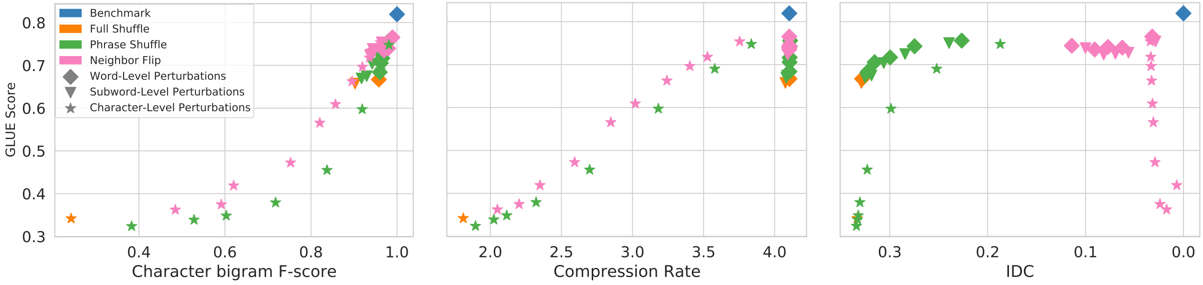
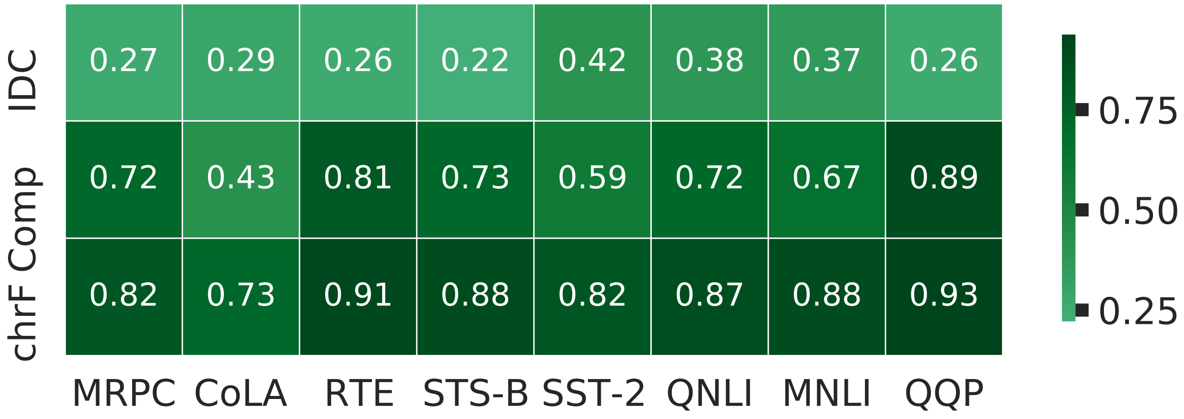


Figure 9: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of PT BART-Base model tested on the perturbed data.



	MRPC	CoLA	RTE	STS-B	SST-2	QNLI	MNLI	QQP
IDC	0.27	0.29	0.26	0.22	0.42	0.38	0.37	0.26
Comp	0.72	0.43	0.81	0.73	0.59	0.72	0.67	0.89
chrF	0.82	0.73	0.91	0.88	0.82	0.87	0.88	0.93

Figure 10: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the PT BART model.

C.2 NPT Transformer

The NPT Transformer has many interesting results that warrant additional analysis. In Figure 11, we can observe that no word or subword-level perturbation have any effect on the models performance, which implies that it considers inputs containing the same subwords in any order as equivalent. In other words, it makes not use of the position of inputs. Looking at individual tasks in Figure 12, we further observe that the correlations to the MRPC, CoLA and RTE tasks are all flat. By observing those tasks performance individually in 13, we can see that the low correlation is simply caused by the fact that the model is incapable to obtain above-chance performances on any of the tasks. Adding the results of the NPT Transformer with positional embeddings frozen to 0, in Figure 14 and Figure 15, we can see little difference between the NPT Transformer with and without positional embedding.

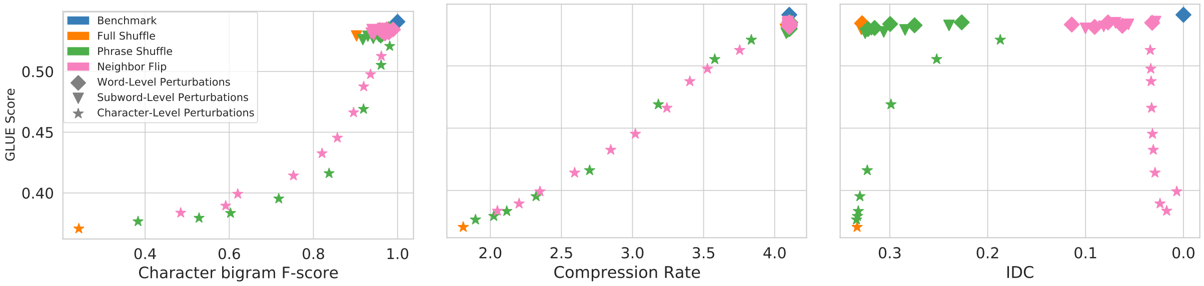
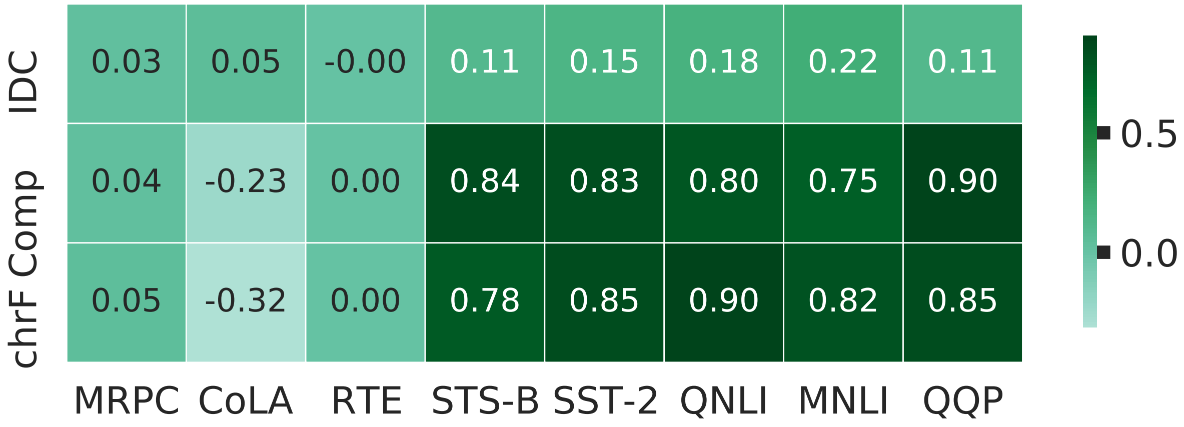
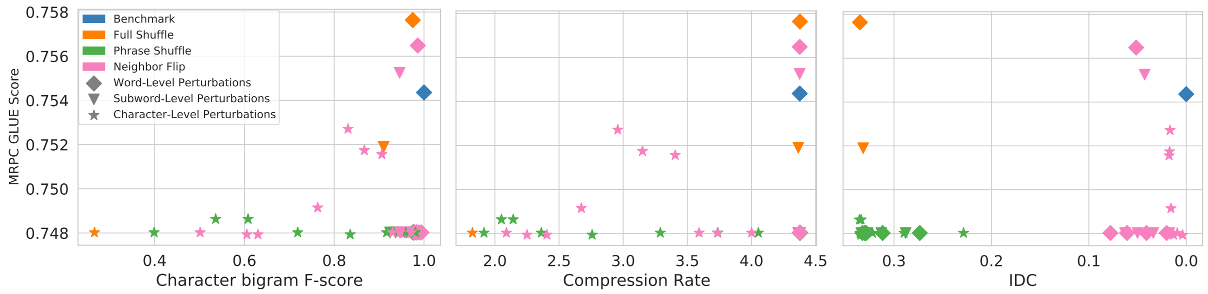


Figure 11: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of NPT Transformer model tested on the perturbed data. The model does not seem to consider the position of tokens which explains why word and subword-level perturbation do not seem to affect the performances.

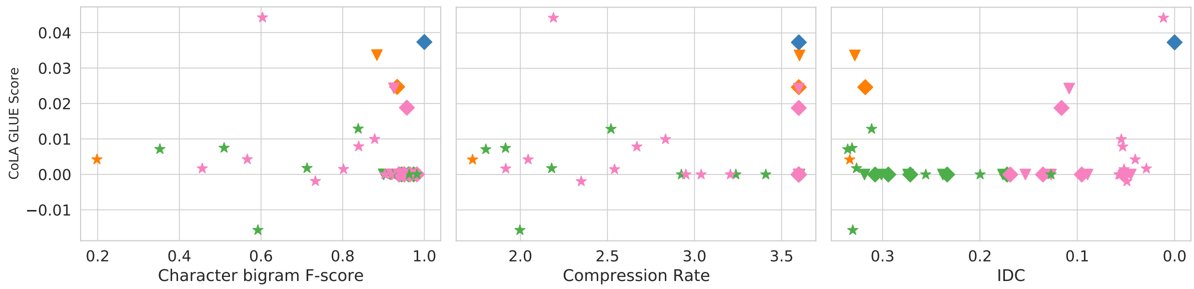


	MRPC	CoLA	RTE	STS-B	SST-2	QNLI	MNLI	QQP
IDC	0.03	0.05	-0.00	0.11	0.15	0.18	0.22	0.11
Comp	0.04	-0.23	0.00	0.84	0.83	0.80	0.75	0.90
chrF	0.05	-0.32	0.00	0.78	0.85	0.90	0.82	0.85

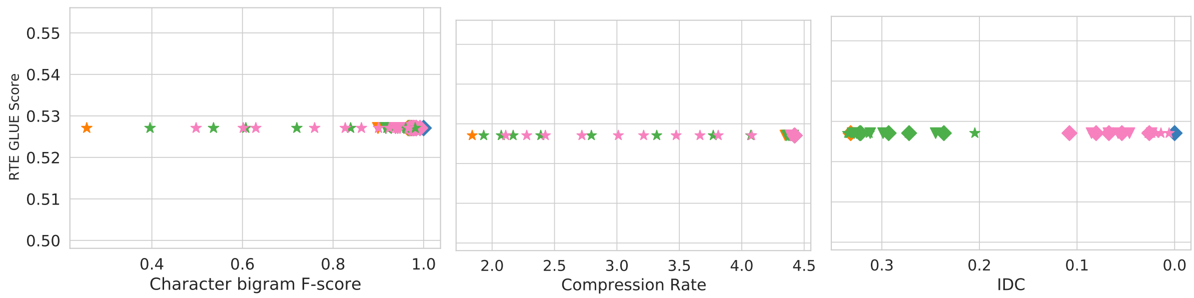
Figure 12: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the NPT Transformer model. The model obtains a static chance score on the RTE task and extremely low scores on the MRPC and CoLA tasks which explains the strange correlations. Those three tasks have seen the greatest improvement on the GLUE benchmark from the introduction of PT models. Those are also the three smallest tasks in the GLUE benchmark lending credence to the idea that positional embeddings are data hungry.



(a) NPT Transformer MRPC



(b) NPT Transformer CoLA



(c) NPT Transformer RTE

Figure 13: Plotted are the offending task for the strangeness in the NPT Transformer correlation. Those tasks seem to rely on the position of inputs more then other tasks which would explain the comparatively poor performance of the NPT Transformer.

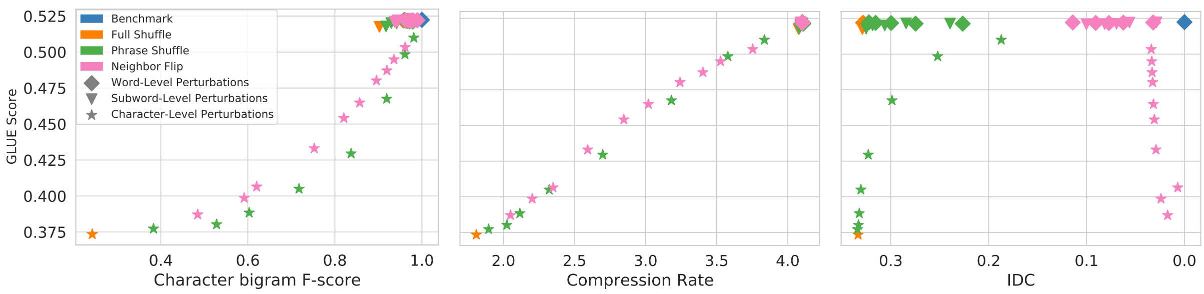


Figure 14: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of NPT Transformer with positional embeddings frozen at 0. We observe very similar results to the NPT Transformers with positional embeddings.

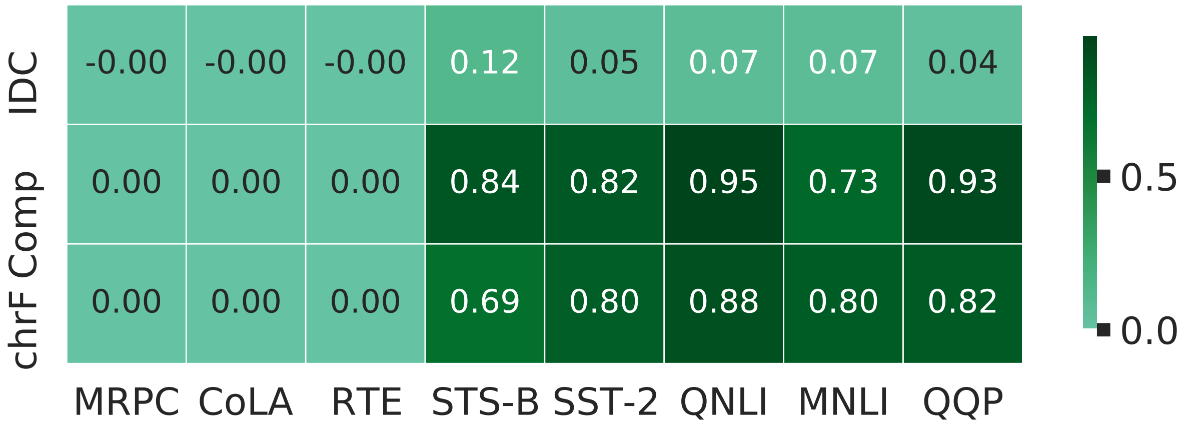


Figure 15: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of with the NPT Transformer with positional embeddings frozen at 0. We observe very similar results to the NPT Transformers with positional embeddings.

C.3 PT CharBERT

The PT CharBERT seem roughly inline with the other PT models, with generally more importance to the chrF-2 and somewhat less importance to the compression rate.

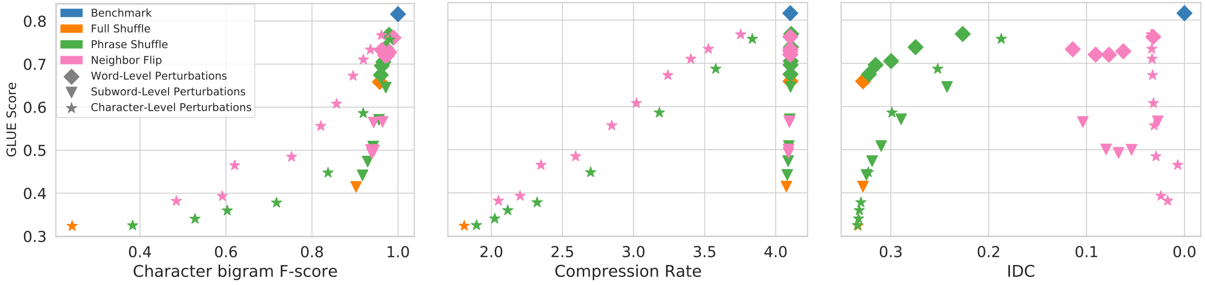


Figure 16: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of PT CharBERT model tested on the perturbed data.

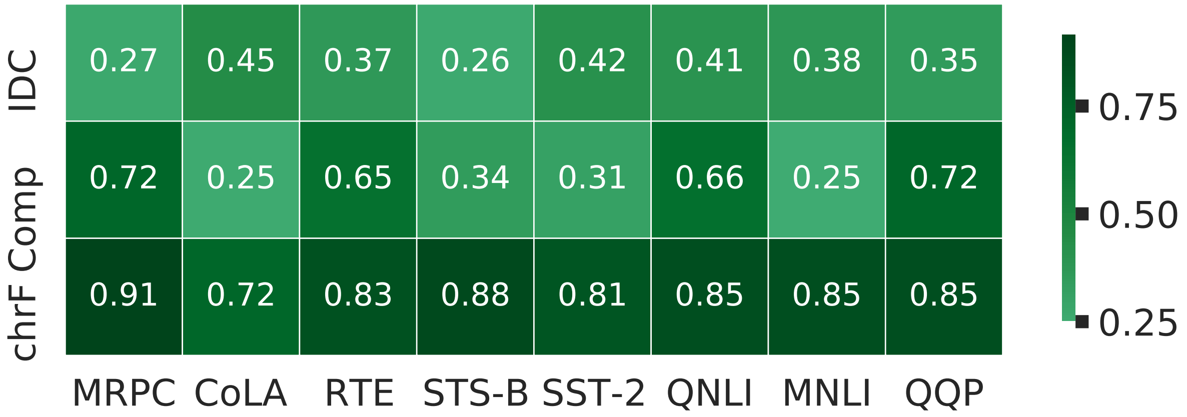


Figure 17: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the PT CharBERT model.

C.4 ConvNet

The ConvNet is inline with other models, with the exception that it fails to obtain any kind of performance on the RTE task.

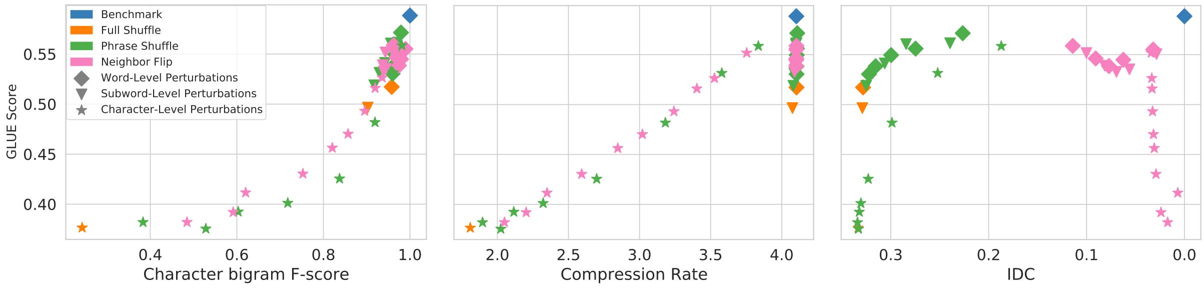


Figure 18: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of ConvNet model tested on the perturbed data.

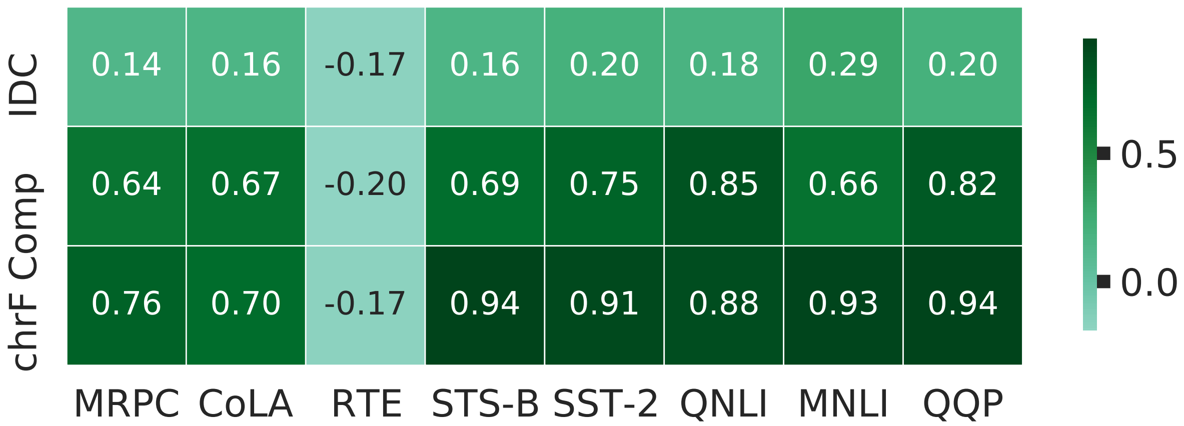


Figure 19: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the ConvNet model. Much like the NPT Transformer, it is unable to obtain above chance-level on the RTE task.

C.5 BiLSTM

The BiLSTM is inline with other models performances.

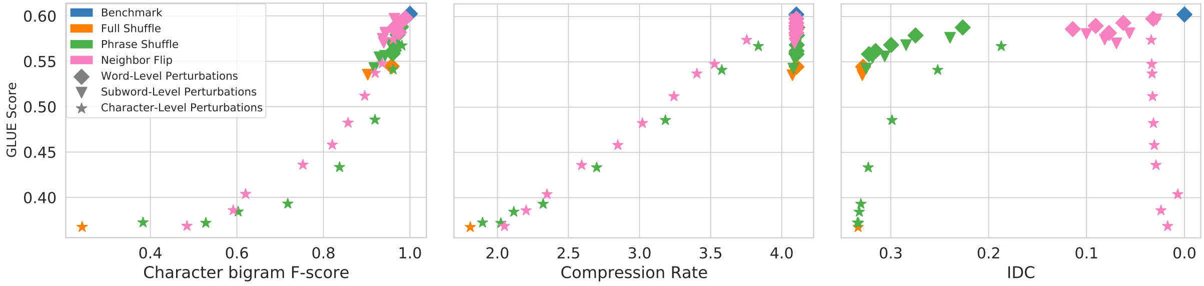


Figure 20: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of BiLSTM model tested on the perturbed data.

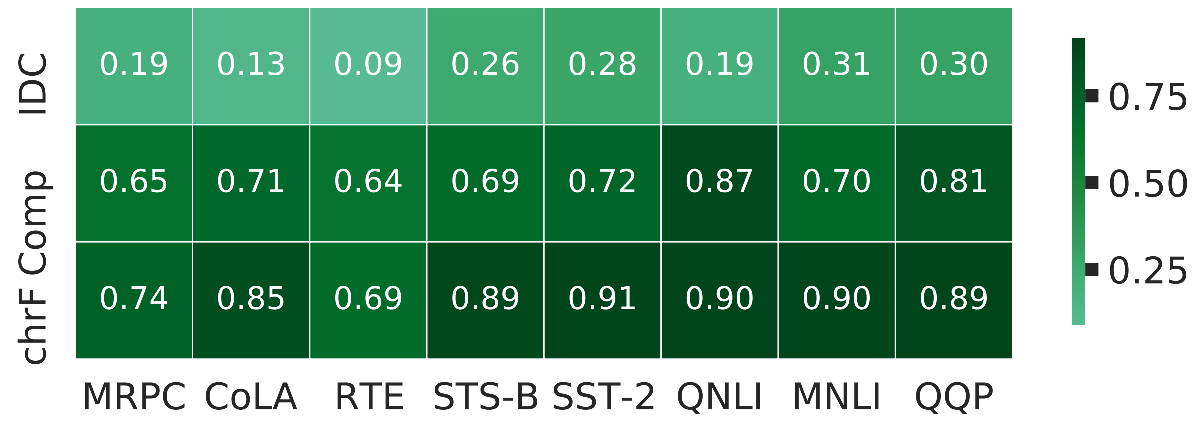


Figure 21: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the BiLSTM model.

C.6 ConvNet Character Embeddings

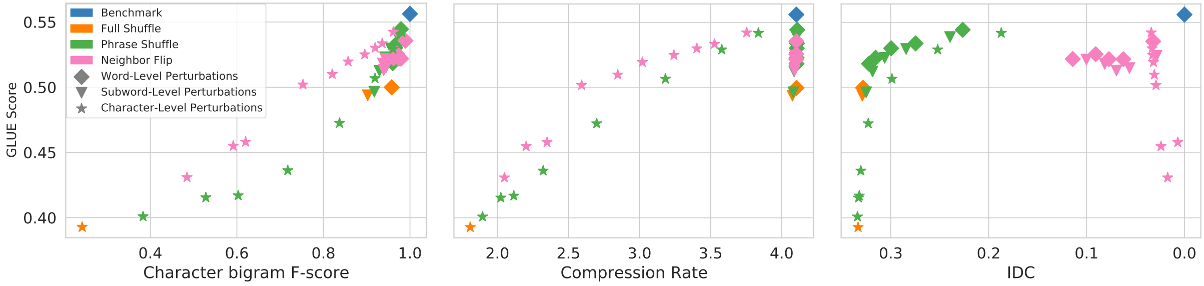


Figure 22: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of BiLSTM model tested on the perturbed data.

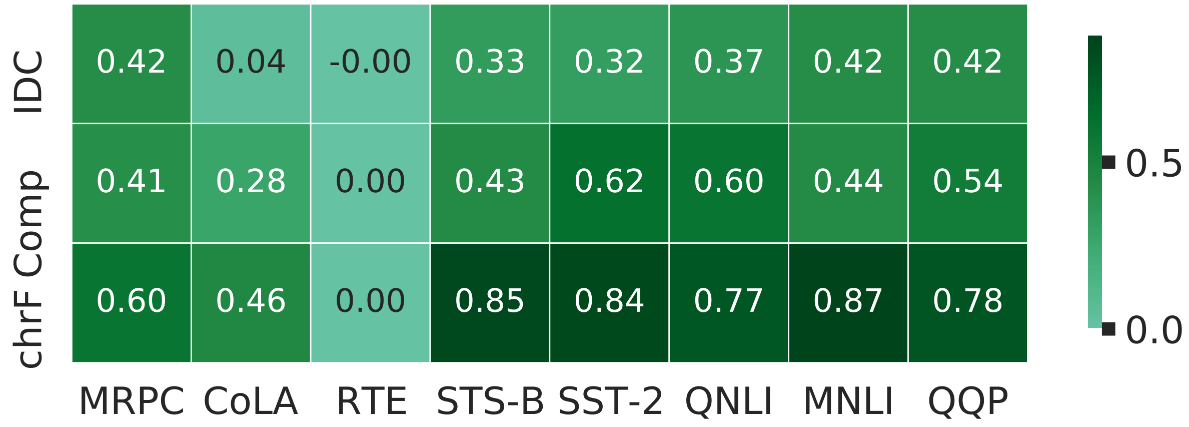


Figure 23: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the BiLSTM model.

C.7 BiLSTM with Character Embeddings

The BiLSTM with Character Embeddings results seem roughly inline with the other models, with some failures on the CoLA, MRPC and RTE tasks.

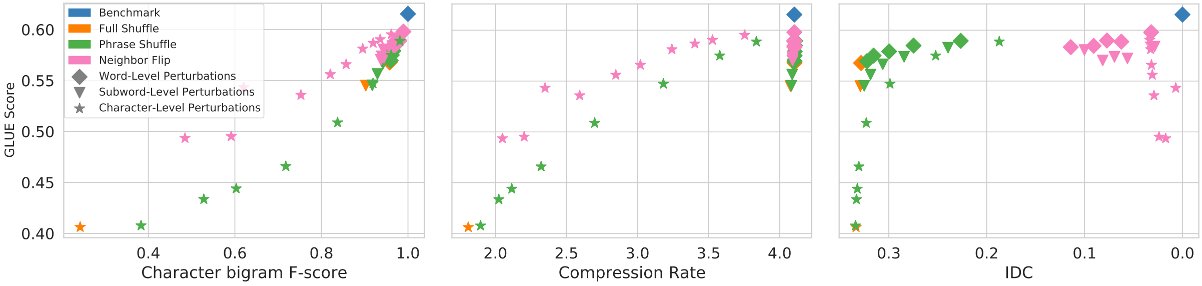
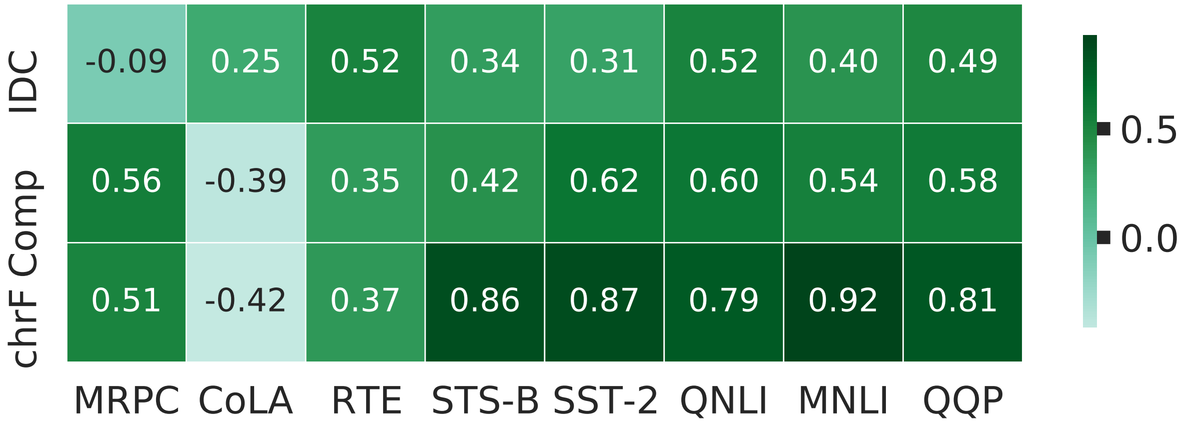


Figure 24: Plotted are the relations between the different choices of metrics measuring the amount of perturbation and the performance of BiLSTM with character embeddings model tested on the perturbed data.



	MRPC	CoLA	RTE	STS-B	SST-2	QNLI	MNLI	QQP
IDC	-0.09	0.25	0.52	0.34	0.31	0.52	0.40	0.49
Comp	0.56	-0.39	0.35	0.42	0.62	0.60	0.54	0.58
chrF	0.51	-0.42	0.37	0.86	0.87	0.79	0.92	0.81

Figure 25: Rank correlation matrix between perturbations measured by different metrics and the performance on the different GLUE tasks of the BiLSTM with character embeddings model. In this case, the model struggles on the RTE and CoLA task.